



L I N E OTM

The Embedded Linux[®] Solutions CompanyTM

Good Fork, Bad Fork

Examining the Limits of Open Source Software in the Embedded Market

WWW.LINEO.COM

Tim Bird

Chief Technology Officer
www.lineo.com

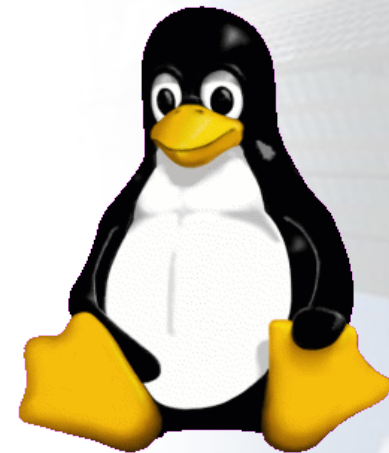


Start with 2 Definitions

- Definition of open source
 - What are it's key attributes
- Definition of network effects
 - Importance of network effects for open source software

What is Open Source Software?

- Examples
 - Linux
 - Apache
 - gcc (GNU compiler)



Key Attributes of Open Source Software

- Access to the source code
- Freedom to make modifications AND distribute them
 - (free = freedom : think free speech, not free beer)
- Licenses that provide these attributes
- Availability of source is NOT enough



Source Availability != Open Source

- QNX now has source availability
- For lots of money, you can buy source code to VxWorks
- Microsoft may ship Windows CE source code
- But that's NOT Open Source



Key Attributes of Open Source Software

- Communities develop
- The "Linux community"
- This generates "network effects"

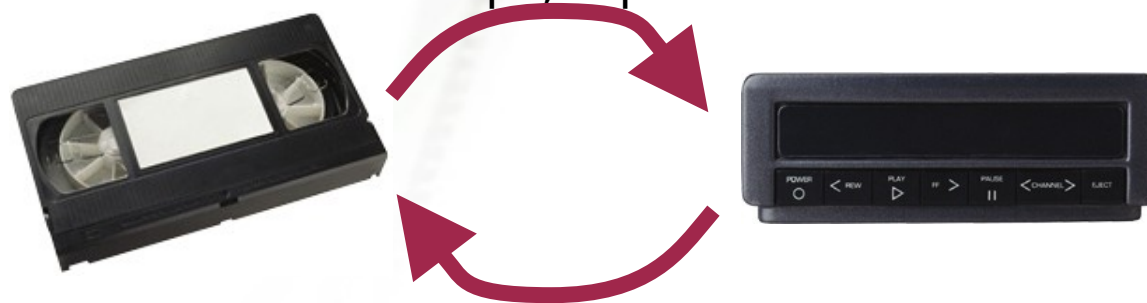
What are “Network Effects”?

- When the value of something increases with the number of items
- Classic example: the telephone
 - Two phones have limited value
 - Whole network of phones gives each one its value

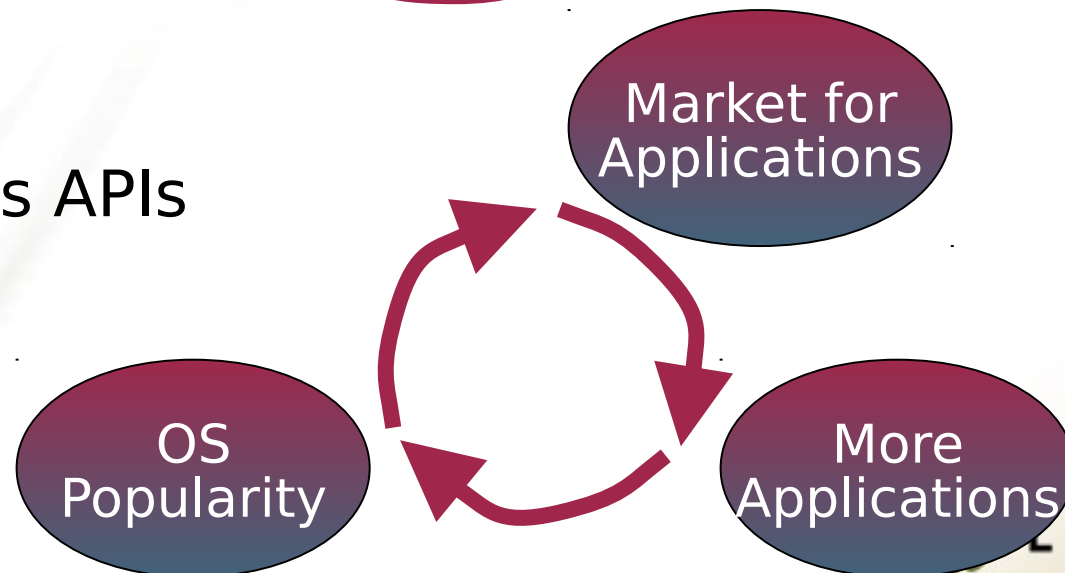


Other “Network Effect” Examples

- Classic example: VHS videocassette tapes
 - Once a standard develops, it pushes other formats out



- Windows APIs



Network Effects and Linux

- Every feature of Linux makes it more valuable to developers
- Every Linux developer makes Linux have more features
- Virtuous cycle



Open Source Network Effects (Business Benefits)

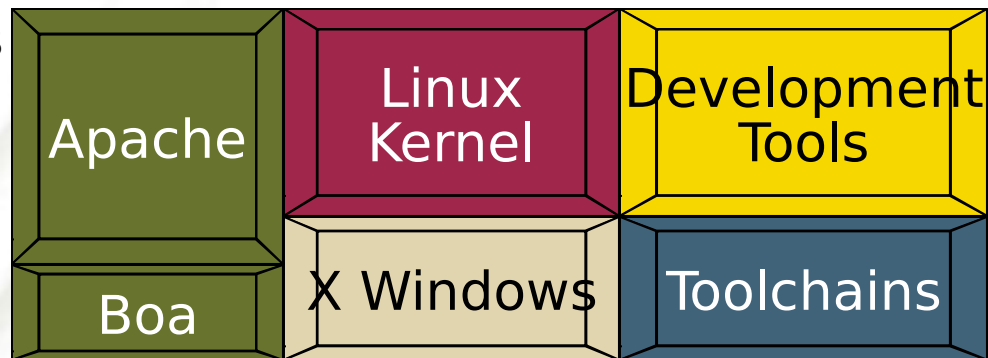
- Popularity
 - Availability of engineering resources
 - Info
 - Skilled manpower
 - Engineer enthusiasm
- Commercial effects
 - Multi-vendor OS
 - Competition to produce rapid development
 - Test organizations



Linux is Not Just One Community

- Separate communities for networking, file systems, Web servers, graphic layers, desktops, etc.
- Size of the community matters
- Some communities are subject to “quantum” effects

– Loss



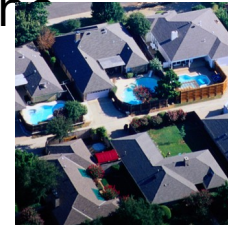
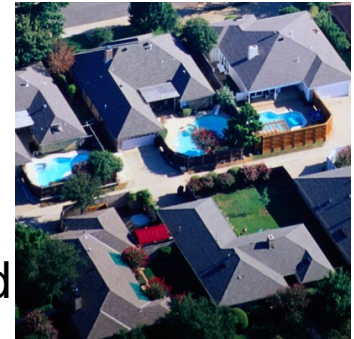
The Problem of Forks

WWW.LINEO.COM



What is a Fork?

- A fork is when the code splits
 - Multiple copies of the same code
- Can occur for many reasons
 - Experimentation with new algorithms
 - Political conflict between groups or individuals
 - Commercial interest in protecting a benefit
 - Adapting an algorithm to a new niche



Historical UNIX Forks

- Each hardware vendor had their own version of UNIX
- Vendors all competed
- Standards developed slowly
 - “The beauty of UNIX standards is that there are so many to choose from.”
- POSIX 1003 = 1003 API's!
 - Duplicates like `fcntl` and `flock`

XENIX



SvOutPlaceObject



UnixWare

ULTRIX

IRIX



HP-UX



LINEO™

Key Points About UNIX Forks

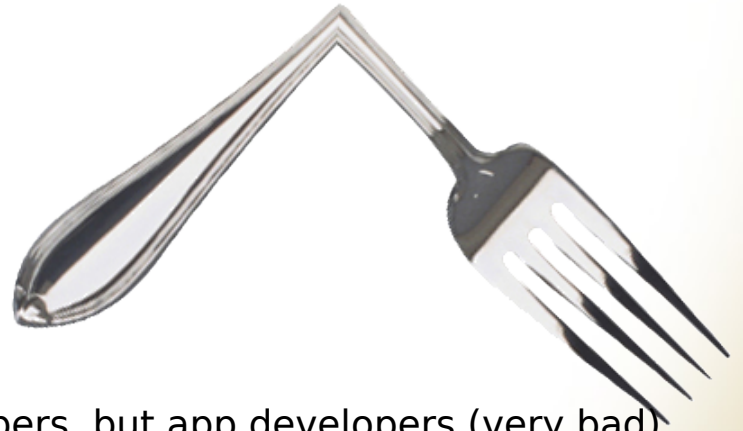
- Each UNIX version was too small to support a complete community of developers
- Overhead of supporting all UNIX versions was too much for developers

UNIX



Bad Forks

- “Overlapping Fork”
 - Divide network (split market, add to overhead)
 - Overlapping forks are prevented in the main tree by GPL
- Examples:
 - UNIX “wars” of the 1980s
 - vi vs. emacs
 - KDE vs. GNOME
 - Divides attention of developers
 - Divides not just desktop developers, but app developers (very bad)
 - Target same audience
 - Differences in:
 - Programming style
 - Geography of developers



Good Forks

- “Specialization” forks
 - Take Linux where it couldn’t go before
 - Create ‘new’ communities
 - Still leverage some value from old community



Good Fork Examples: uClinux

- Currently kernel version 2.0.38
- MMU-less
- Uses many Linux facilities
 - Some programs are hard to port
 - Requires backporting of some drivers
- Has its own community
- Is decidedly non-overlapping

LINEO™
μClinux™



Good/Bad Fork Example: Real-Time

- Different realtime approaches
 - Interrupt abstraction with separate scheduler
 - Hard realtime in kernel space
 - RTAI and RTLinux projects
 - Projects overlap in implementation
 - Now have a common API
 - Preemption improvement
 - Soft realtime in user space
 - Attacks a different market need than interrupt abstraction approach



Linus Says:

"Fragmentation is the sort of bogeyman of Unix, but fragmentation is often good. Most of the things about fragmentation I like. You want to have a market where everybody gets to do their own thing and where one entity doesn't control it."

Torvalds also distinguished between good and bad fragmentation, bemoaning heavily the fact that bad software fragmentation -- fueled by politics and needless feature differentiation -- still exists in the market.

Good fragmentation, on the other hand, makes it possible for Linux to run in both supercomputers and refrigerators, Torvalds said. "The key is modularity. You don't solve every problem with one huge operating system"

(ZDNet Interview - February, 2000)

Linus Says (2):

Q: Do you think Linux will avoid following the same "fragmentation" route that killed UNIX? Why and how?

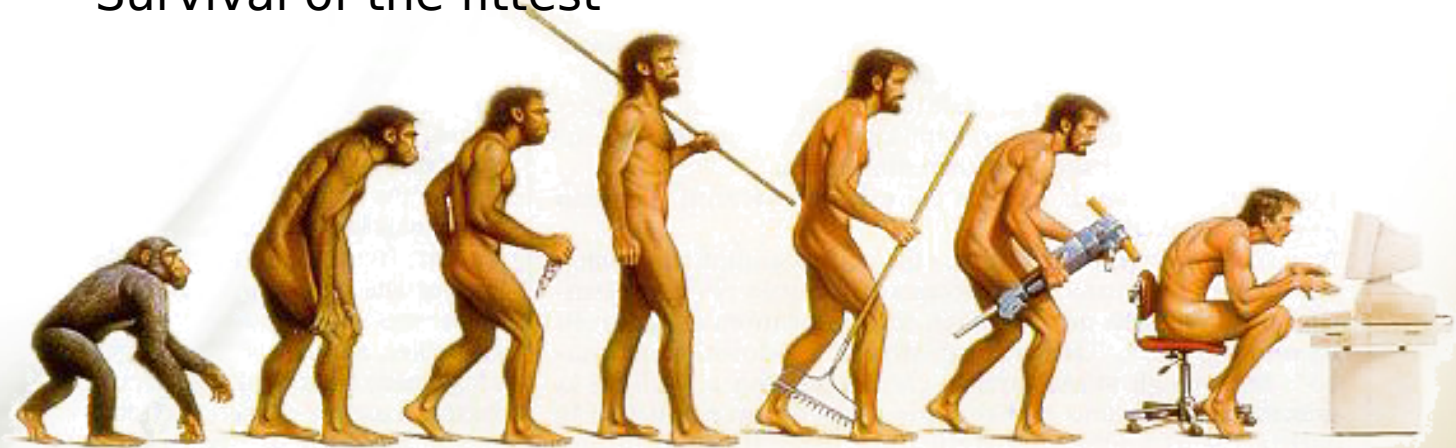
A: It will, and it won't. Let me explain. What made the UNIX fragmentation so bad was that it was an "overlapped" fragmentation - pretty much every single UNIX vendor went after the same market, which meant that every fragment really wanted to do the same things, but because of the lack of openness everybody ended up spending inordinate amounts of energy to re-invent the wheel that somebody else had already done.

And that fragmentation will not happen under Linux. Simply because people do not have to write their own versions of the same thing. They may end up *improving* on somebody else's version, or writing a new and improved thing that everybody else can use too, and that's all to the good. But we won't have people fighting on basic simple infrastructure issues, simply because all the real infrastructure is under a license that really forces the different companies to share.

But what will happen is that the "market" fragments, as opposed to the technology. Which is good and proper. You'll have different Linux companies going after different markets, and having different priorities. Which means that you won't have any one particular "microsoft of Linux": you'll have dominant players, but they'll be dominant in the areas that they are good at. And nobody ever ends up being the best at everything.

Evolution Metaphor

- Evolution: mutation and natural selection
- Mutation = fork
- Natural selection = domination of superior version
 - “Survival of the fittest”



Evolution in Software

- Bad ideas go away...people don't work on them



Featured in The Best of
Dave Karger's Holiday Gift
Guide for the 1996

Miami Herald:

®Larvae are actual
beet larvae that
supposed

Ein offenes Betriebssystem
kann schon mal mutieren. Bei
Windows 2000 hingegen gibt
es alle Services und Dienste
aus einer Hand. Das spart Zeit
und somit wirklich Geld. Mehr
Infos unter [www.microsoft.com/
germany/windows2000](http://www.microsoft.com/germany/windows2000)

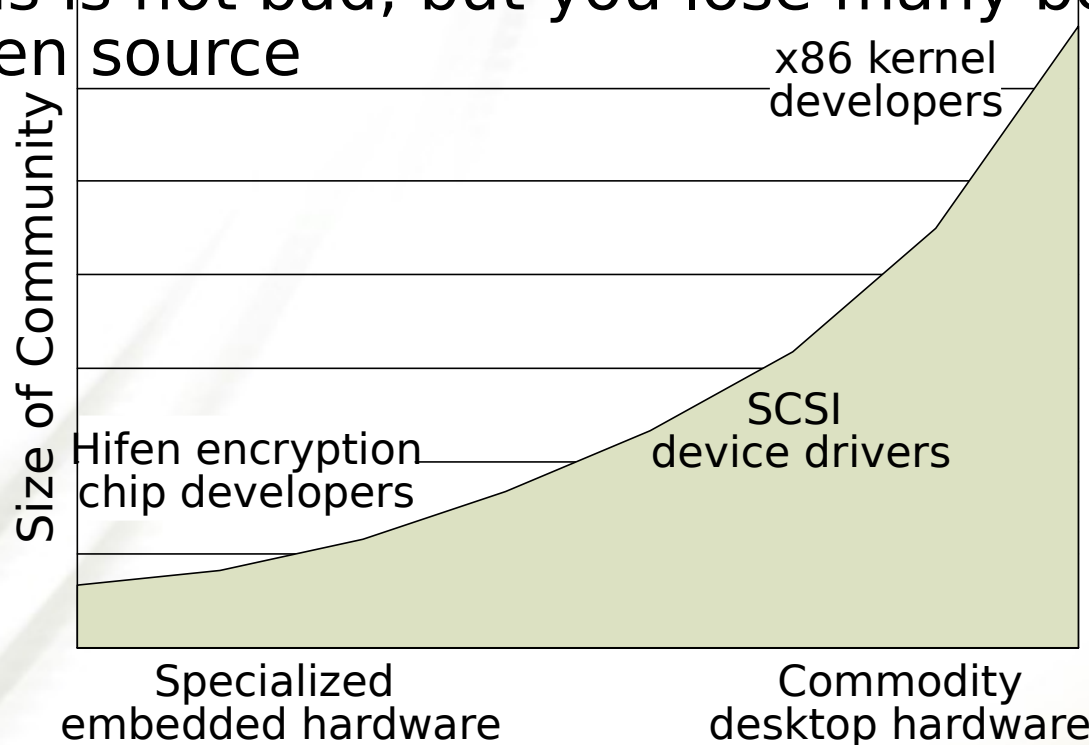
Microsoft®



ein offenes betriebssystem hat nicht nur vorteile

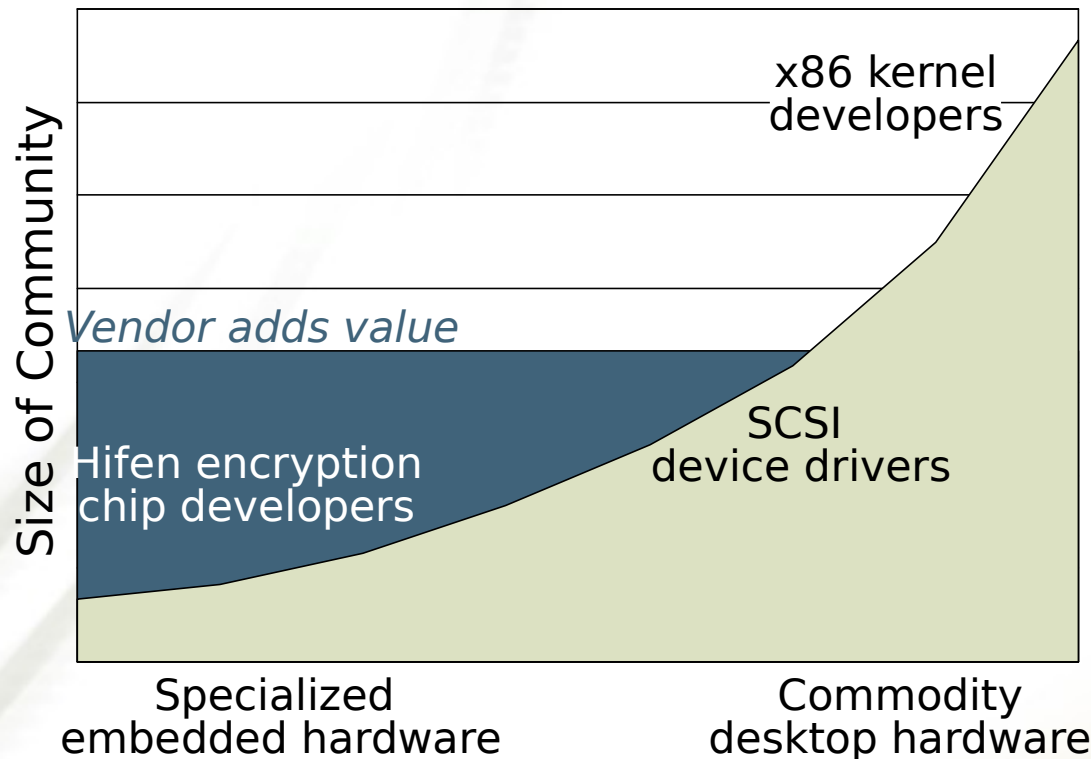
Embedded World Makes for Small Communities

- Communities are small (if they exist at all), because regular developers don't have the hardware
- This is not bad, but you lose many benefits of open source



How to Solve this Problem

- Stay close to the desktop and commodity markets
- Pay money where necessary



Commercial Developers can:

- Make up for loss of community
- Testing (complements or replaces peer review)
- Development
- Training
- Build a community



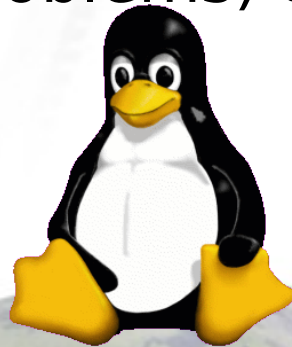
Conclusion

- Not all forks are bad
- Identify the forks, embrace and support the good, avoid the bad
- Embedded market leads to specialization which leads to smaller communities and reduced beneficial network effects
- Commercial developers can fill the gap



Linux will Still Dominate the Embedded Market

- Linux is not free (but then, nothing in life is free)
- Linux has so many benefits (even with the specialization problems) that it WILL win





L I N E OTM

The Embedded Linux[®] Solutions CompanyTM